THE NATIONAL INSTITUTE OF ENGINEERING

MYSORE – 8

(Autonomous Institution under VTU)

B.E (Computer Science and Engineering)

& CSE (AI&ML)

Detailed Syllabus – I Year

1st & 2nd Semester

Department of Computer Science and Engineering

The National Institute of Engineering

Scheme of Teaching & Examination - 2025 scheme

B.E. 2025 Admission

I Semester - Physics Cycle

Sl. No	Course and Course Code		Course Title	TD/PSB	Teaching Hours/Week				Examination				Credit
					Theory Lecture	Tutorial	Practical/ Drawing	SDA	Duration in hours	CIE Marks	SEE Marks	Total Marks	
1	ASC	1BMATS101	Calculus and Linear Algebra: CSE Stream	Maths	3	2	0		3	50	50	100	4
2	ASC (IC)	1BPHYS102	Quantum Physics and Applications (CSE stream)	Physics	3	0	2		3	50	50	100	4
3	ESC	1BCEDS103	Computer-Aided Engineering Drawing for CSE stream	ME Dept	2	0	2		3	50	50	100	3
4	ESC	1BXXX104X	Engineering Science Course-I	Respective Engg dept	3	0	0		3	50	50	100	3
5	PSC	1BEIT105	Programming in C	CSE	3	0	0		3	50	50	100	3
	OR												
	PLC	1BPLC105B	Python Programming										
6	AEC (NCMC)	1BSKS106	Soft Skills	Humanities	1	0	0		-	100	-	100	PP
7	PSCL	1BPOPL107	C Programming Lab	CSE	0	0	2		2	50	50	100	1
	OR												
	PLCL	1BPOPL107B	Python Programming Lab										
8	AEC/SDC	1BIDTL158	Innovation and Design Thinking Lab (Project-based learning)	CSE	0	0	2		2	50	50	100	1
9	HSMS	1BKSK109(BKSK107)/ 1BKBK109(BKBK107)	Samskrutika Kannada/ Balake Kannada	Humanities	1	0	0		1	50	50	100	1
				Total	**16**	**02**	**08**		**20**	**500**	**400**	**900**	**20**

The National Institute of Engineering

Scheme of Teaching & Examination - 2025 scheme

B.E. 2025 Admission

I Semester - Chemistry Cycle

Sl. No	Course and Course Code		Course Title	TD/PSB	Teaching Hours/Week				Examination				Credit
					Theory Lecture	Tutorial	Practical/ Drawing	SDA	Duration in hours	CIE Marks	SEE Marks	Total Marks	
1	ASC	1BMATS101	Calculus and Linear Algebra: CSE Stream	Maths	3	2	0		3	50	50	100	4
2	ASC (IC)	1BCHES102	Applied Chemistry for Smart Systems (CSE stream)	Chemistry	3	0	2		3	50	50	100	4
3	ETC	1BAIA103	Introduction to AI and Applications	Any Dept	3	0	0		3	50	50	100	3
4	ESC	1BESC104x	Engineering Science Course-I	Respective Engg dept	3	0	0		3	50	50	100	3
5	PSC	1BEIT105	Programming in C	CSE	3	0	0		3	50	50	100	3
	OR												
	PLC	1BPLC105B	Python Programming										
6	AEC	1BENGL106	Communication Skills	Humanities	1	0	0		2	50	50	100	01
7	AEC (NCMC)	1BICO107	Indian Constitution & Engineering Ethics	Humanities	1	0	0		-	100	-	100	PP
8	AEC/SDC	1BIDTL158	Innovation and Design Thinking Lab (Project-based learning)	Any Dept	0	0	2		2	50	50	100	1
9	PSCL	1BPOPL109	C Programming Lab	CSE	0	0	2		2	50	50	100	1
	OR												
	PLCL	1BPOPL109B	Python Programming Lab										
			Total		**17**	**02**	**06**		**2**	**500**	**400**	**900**	**20**

The National Institute of Engineering

Scheme of Teaching & Examination - 2025 scheme

B.E. 2025 Admission

II Semester - Physics Cycle

Sl. No	Course and Course Code		Course Title	TD/PSB	Teaching Hours/Week				Examination				Credit
					Theory Lecture	Tutorial	Practical/ Drawing	SDA	Duration in hours	CIE Marks	SEE Marks	Total Marks	
1	ASC	1BMATS201	Numerical Methods: CSE Stream	Maths	3	2	0	0	3	50	50	100	4
2	ASC (IC)	1BPHYS202	Quantum Physics and Applications (CSE stream)	Physics	3	0	2		3	50	50	100	4
3	ESC	1BCEDS203	Computer-Aided Engineering Drawing for CSE stream	ME Dept	2	0	2		3	50	50	100	3
4	ESC	1BXXX204X	Engineering Science Course-I	Respective Engg dept	3	0	0		3	50	50	100	3
5	PSC	1BEIT205	Programming in C	CSE	3	0	0		3	50	50	100	3
	OR												
	PLC	1BPLC205B	Python Programming										
6	AEC (NCMC)	1BSKS206	Soft Skills	Humanities	1	0	0		-	100	-	100	PP
7	PSCL	1BPOPL207	C Programming Lab	CSE	0	0	2		2	50	50	100	1
	OR												
	PLCL	1BPOPL207B	Python Programming Lab										
8	AEC/SDC	1BIDTL258	Interdisciplinary Project-Based Learning	Combination of Departments	0	0	0	02	2	50	50	100	1
9	HSMC	1BKSK209(BKSK107)/ 1BKBK209(BKBK107)	Samskrutika Kannada/ Balake Kannada	Humanities	1	0	0		1	50	50	100	1
				Total	**16**	**02**	**06**	**02**	**20**	**500**	**400**	**900**	**20**

The National Institute of Engineering

Scheme of Teaching & Examination – 2025 scheme

B.E. 2025 Admission

II Semester - Chemistry Cycle

Sl. No	Course and Course Code		Course Title	TD/PSB	Teaching Hours/Week				Examination				Credit
					Theory Lecture	Tutorial	Practical / Drawing	SDA	Duration in hours	CIE Marks	SEE Marks	Total Marks	
1	ASC	1BMATS201	Numerical Methods: CSE Stream	Maths	3	2	0		3	50	50	100	4
2	ASC (IC)	1BCHES202	Applied Chemistry for Smart Systems (CSE)	Chemistry	3	0	2		3	50	50	100	4
3	ETC	1BAIA203	Introduction to AI and Applications	Mechanical Engg	3	0	0		3	50	50	100	3
4	ESC	1BESC204X	Engineering Science Course-II	Respective Engg dept	3	0	0		3	50	50	100	3
5	PSC	1BEIT205	Programming in C	CSE	3	0	0		3	50	50	100	3
	OR												
	PLC	1BPLC205B	Python Programming										
6	AEC	1BENGL206	Communication Skills	Humanities	1	0	0		2	50	50	100	1
7	AEC (NCMC)	1BICO207	Indian Constitution & Engineering Ethics	Humanities	1	0	0		1	100	0	100	PP
8	AEC/SDC	1BPRJ258	Interdisciplinary Project-Based Learning	Respective Dept (Multiple Dept)	0	0	0	2	2	50	50	100	1
9	PSCL	1BPOPL209	C Programming Lab	CSE	0	0	2		2	50	50	100	1
	OR												
	PLCL	1BPOPL209B	Python Programming Lab										
			Total		**17**	**2**	**4**	**2**	**22**	**500**	**400**	**900**	**20**

Engineering Science Courses (ESC)				
Code	**Course Title**	**L**	**T**	**P**
1BESC104A/204A	Building Science and Machines	3	0	0
1BESC104B/204B	Introduction to Electrical Engineering	3	0	0
1BESC104C/204C	Introduction to Electronics & Communication Engineering	3	0	0
1BESC104D/204D	Introduction to Mechanical Engineering	3	0	0
1BESC104E/204E	Essentials of Information Technology *	3	0	0

*CSE students cannot register for this course.

Programming Language Courses (PLC)				
Code	**Course Title**	**L**	**T**	**P**
1BPLC105E/205E	Introduction to C Programming (for non-IT programmes)	3	0	0
1BPLC105B/205B	Python Programming (For CSE and allied programmes)	3	0	0

Programming Language Course Lab (PLCL)				
Code	**Course Title**	**L**	**T**	**P**
1BPOPL107E/109E/207E/209E	Programming in C Lab (for non-IT programmes)	3	0	0
1BPOPL107B/109B/207B/209B	Python Programming (For CSE and allied programmes)	3	0	0

Code: 1BEIT105/1BEIT205 **Course: Programming in C**

Credits: 3 **L: T:P – 3:0:0**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	NIL
Learning objectives	To acquire fundamental programming concepts, methodologies, and structures that are essential for developing efficient and reliable C programs.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Demonstrate fundamental concepts and language constructs of C programming.	Understand
CO2	Develop programs using control structures, arrays and user-defined functions for complex computational problems.	Apply
CO3	Examine the usage of structures, unions and enumerations to model simple real-world scenarios.	Analyze
CO4	Choose suitable data types and language constructs to solve a given computational or real-world problem.	Evaluate

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11		PSO1	PSO2
CO1	3	2	1	1	1	-	-	1	1	-	1		2	1
CO2	3	3	3	3	2	-	-	2	1	-	2		3	2
CO3	3	3	3	3	2	-	-	2	2	-	2		3	2
CO4	3	3	3	3	2	-	-	2	2	-	2		3	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

No.	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module 1				
1.1	**Introduction to Computing:** Computer languages, Creating and Running Programs, System Development.	1		
1.2	A Brief History of C, C Is a Middle-Level Language,	1		
1.3	C Is a Structured Language; C Is a Programmer's Language	1		
1.4	Compilers Vs. Interpreters, The Form of a C Program, The Library and Linking, Separate Compilation,	1		
1.5	Compiling a C Program, C's Memory Map. The Basic Data Types Modifying the Basic Types	1		
1.6	Identifier Names, Variables, The Four C Scopes	1		
1.7	Type Qualifiers, Storage Class Specifiers, Variable Initializations,	1		
1.8	Constants, Operators, Expressions.	1		
Module 2				
2.1	Console I/O: Reading and Writing Characters,	1		
2.2	Reading and Writing Strings	1		
2.3	Formatted Console I/O, printf(), scanf().	1		
2.4	Statements: True and False in C,	1		
2.5	Selection Statements	1		
2.6	Iteration Statements	1		
2.7	Jump Statements, Expression Statements,	1		
2.8	Block Statements.	1		
Module 3				
3.1	Arrays and Strings: Single-Dimension Arrays,	1		
3.2	Generating a Pointer to an Array, Passing Single-Dimension	1		
3.3	Arrays to Functions, Strings, Two-Dimensional Arrays,	1		
3.4	Multidimensional Arrays, Array Initialization, Variable - Length Arrays.	1		
3.5	Pointers: What Are Pointers?, Pointer Variables,	1		
3.6	The Pointer Operators, Pointer Expressions,	1		

3.7	Pointers and Arrays, Multiple Indirection,	1		
3.8	Initializing Pointers.	1		
Module 4				
4.1	Functions: The General Form of a Function,	1		
4.2	Understanding the Scope of a Function, Function Arguments,	1		
4.3	argc and argv—Arguments to main (),	1		
4.4	The return Statement, What Does main () Return?	1		
4.5	Recursion, Function Prototypes,	1		
4.6	Declaring Variable Length Parameter Declarations,	1		
4.7	The inline Keyword. Pointers (Contd…): Pointers to Functions,	1		
4.8	C's Dynamic Allocation Functions.	1		
Module 5				
5.1	Structures, Arrays of Structures	1		
5.2	Passing Structure to Functions,	1		
5.3	Structure Pointers,	1		
5.4	Arrays and Structures within Structures	1		
5.5	Unions, Bit-Fields	1		
5.6	Enumerations	1		
5.7	Using size of to Ensure Portability, typedef	2		
	Total No. of Lecture Hours	40	-	-
	Total No. of Tutorial Hours		00	-
	Total No. of Practical Hours			00

Text book:

1. Schildt, Herbert." C the complete reference",4th Edition, Mc Graw Hill.
2. Hassa A fyouni, Behrouz A. Forouzan. “A Structured Programming Approachin”,4th Edition, Cengage.

Reference Book:

1. Brian W. Kernighan and Dennis M.Ritchie, The 'C' Programming Language,2nd Edition, Prentice Hall of India.
2. Reema Thareja , Programming in C,3rd Edition, Oxford University Press, 2023.

Online Resources:

1. elearning.vtu.ac.in/econtent/courses/video/BS/15PCD23.html
2. Introduction to Programming in C [https://onlinecourses.nptel.ac.in/noc23_cs02/preview]
3. C for Everyone: Programming Fundamentals [https://www.coursera.org/learn/c-for-everyone]
4. Computer Programming Virtual Lab[https://cse02-iiith.vlabs.ac.in/exp/pointers/]
5. C Programming: The ultimate way to learn the fundamentals of the C language [https://www.pdfdrive.com/c-programming-the-ultimate-way-to-learn-the-fundamentals-of-the-c-language-e187584209.html]
6. C Programming: The Complete Reference [https://viden.io/knowledge/programming-in-c-language/attachment/28313/c-the-complete-reference-herbert-schildt-4th-edition-pdf/preview]
7. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013843237039374336345 17_shared/overview

Code: 1BPLC105B/1BPLC205 **Course: Python Programming**

Credits: 3 **L:T:P - 3:0:0**

SEE: 50 Marks **CIE:50 Marks**

SEE Hours: 3 **Total Marks:100**

Prerequisites if any	C, C++
Learning objectives	To gain insights into Python language syntax & semantics and use lists, tuples, built-in functions & object-oriented programming concepts to solve problems.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Interpret the syntax and semantics of simple code snippets of Python programming	Understand
CO2	Develop programs for primitive data structures,string processing and file organization in Python	Apply
CO3	Analyze the given python programs to identify errors and performance bottleneck	Analyze
CO4	Implement Object Oriented Programming principles in Python programs	Apply

Mapping with Pos and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	1	1	1	1	2	1	1	1	1	3	3	2
CO2	3	3	3	2	3	3	1	1	1	1	3	3	2
CO3	3	3	3	2	3	2	1	1	1	1	3	3	2
CO4	2	3	3	2	3	3	1	1	1	1	3	3	2

Mapping Strength: Strong–3 Medium–2 Low–1

Course Structure

No.	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module–1			
1.1	**The way of the program:** The Python programming language, what is a program? What is debugging? Syntax errors, Runtime errors, Semantic errors, Experimental debugging.	2	-	-
1.2	**Variables, Expressions and Statements:** Values and data types, Variables, Variable names and keywords, Statements, Evaluating expressions, Operators and operands, Type converter functions, Order of operations, Operations on strings, Input, Composition, The modulus operator.	2	-	-
1.3	**Iteration:** Assignment, updating variables, the for loop, the while statement, The Collatz 3n + 1 sequence, tables, two-dimensional tables, break statement, continue statement, paired data, Nested Loops for Nested Data.	2	-	-
1.4	**Functions:** Functions with arguments and return values. **Chapters: 1.1-1.7, 2.1-2.12, 3.3, 4.4, 4.5**	2	-	
	Module–2			
2.1	**Strings:** Working with strings as single things, working with the parts of a string, Length, Traversal and the for loop, Slices, String comparison, Strings are immutable, the in and not in operators, A find function, Looping and counting, Optional parameters, The built-in find method, The split method, Cleaning up your strings, The string format method.	2	-	
2.2	**Tuples:** Tuples are used for grouping data, Tuple assignment, Tuples as return values, Composability of Data Structures.	2		
2.3	**Lists:** List values, accessing elements, List length, List membership, List operations, List slices, Lists are mutable, List deletion, Objects and references, Aliasing, cloning lists, Lists and for loops, List parameters, List methods	2		
2.4	Pure functions and modifiers, Functions that produce lists, Strings and lists, list and range, Nested lists, Matrices. **Chapter: 5.1, 5.2, 5.3**	2		

Module 3				
3.1	**Dictionaries:** Dictionary operations, dictionary methods, aliasing and copying.	2		
3.2	**Numpy:** About, Shape, Slicing, masking, Broadcasting, dtype.	2		
3.3	**Files:** About files, writing our first file, reading a file line-at-a-time, turning a file into a list of lines, Reading the whole file at once	3		
3.4	Working with binary files, Directories, fetching something from the Web. **Chapter: 5.4, 6.1-6.5, 7.1-7.8**	1		
Module–4				
4.1	**Modules:** Random numbers, the time module, the math module, creating your own modules, Namespaces, Scope and lookup rules, Attributes and the dot Operator	2		
4.2	Three import statement variants. **Mutable versus immutable and aliasing**	2		
4.3	**Object oriented programming:** Classes and Objects — The Basics, Attributes, Adding methods to our class	2		
4.4	Instances as arguments and parameters, Converting an instance to a string, Instances as return values. **Chapter: 8.1-8.8, 9.1, 11.1**	2		
Module–5				
5.1	**Object oriented programming**: Objects are mutable, Sameness, Copying.	3	-	-
5.2	**Inheritance:** Pure function, Modifiers, Generalization, Operator Overloading, Polymorphism.	3	-	-
5.3	**Exceptions**: Catching Exceptions, Raising your own exceptions. **Chapter: 11.2.2-11.2.4, 11.3.2-11.3.9, 12.1, 12.2**	2	-	
	Total No. of Lecture Hours	40	-	-
	Total No. of Tutorial Hours		00	
	Total No. of Practical Hours			00

Text books:

1. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers- How to think like a computer scientist: learning with python 3. Green Tea Press, Wellesley, Massachusetts,2020 https://media.readthedocs.org/pdf/howtothink/latest/howtothink.pdf

2. Al Sweigart," Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners",2nd Edition, No Starch Press, 2022. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)

Reference Books

1. Kyla McMullen, Elizabeth Matthews and June Jamrich Parsons, Programming with Python, Cengage, 2023.

Web links and Video Lectures (e-Resources):

1. https://www.learnbyexample.org/python/
2. https://www.learnpython.org/
3. https://pythontutor.com/visualize.html#mode=edit

Code: 1BPOPL107/109/207/209 **Course: C Programming Lab**

Credits: 1 **L:T:P –** 0:0:2:0

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	Nil
Learning objectives	To acquire fundamental programming concepts, methodologies, and structures that are essential for developing efficient and reliable C programs.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Develop programs in C to solve computational problems.	Apply
CO2	Implement C language programs for real-world applications.	Analyse
CO3	Build a document consisting of experiment setup, design, implementation and results with inferences.	Create

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11		PSO1	PSO2
CO1	3	3	2	2	2				1	1			3	2
CO2	3	3	3	2	2	1	1		2	2	1		3	3
CO3	2	2	3	2	1			1	2	3	1		2	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

List of Experiments

Sl.No	CO	Experiment
PART – A CONVENTIONAL EXPERIMENTS		
1	CO1	A robot needs to find how far it must travel between two points on a 2D plane. Develop a C program to calculate the straight-line distance between the given coordinates.
2	CO1	Develop a C program that takes a student's marks as input and displays their grade based on the following criteria: 90 and above: Grade A 75 to 89: Grade B 60to74:GradeC 50 to 59: Grade D Below50:GradeF Choose a suitable control structure to implement this logic efficiently.
3	CO1	Develop a C program that takes a unique identification input like PAN Number, AADHAR_Number, APAAR_Id, Driving License, Passport and checks it against a set of stored KYC records. Based on the input, display whether the individual is verified or not. Use an appropriate control structure to handle multiple possible ID matches. Assume all Unique identification are of integer type.
4	CO1	A math app needs to determine the type of roots for a quadratic equation based on user input. Develop a C program to calculate and display the roots based on the given coefficients.
5	CO1	A sensor in a robotic arm needs to calculate the angle of rotation in real-time, but the hardware doesn't support built-in trigonometric functions. Develop a C program to approximate the value of sin(x) using a series expansion method for improved performance.
6	CO1	Develop a C program that accepts a course description string and a keyword from the user. Search whether the keyword exists within the course description using appropriate string functions. If found, display: "Keyword '<keyword>' found in the course description." Otherwise, display: "Keyword '<keyword>' not found in the course description."
7	CO1	Develop a C program that takes marks for three subjects as input. Use a function to check if the student has passed (minimum 40 marks in each subject).Display the average and whether the student passed or failed.
8	CO1	In an ATM system, two account balances need to be swapped temporarily for validation. Develop a C program that accepts two balances and uses a function with pointers to swap them. Display the balances before and after swapping.

		PART–B TYPICAL OPEN-ENDED EXPERIMENTS
1	CO2 CO3	A college library has a digital book shelf system where each book is assigned a unique Book ID. The book shelf is organized in ascending order of Book IDs. Develop a C Program to quickly find whether a book with a specific Book ID is available in the shelf.
2	CO2 CO3	A sports teacher has recorded the scores of students in a 100-meter race. To prepare the result sheet, the teacher wants the scores arranged in descending order (from highest to lowest). Develop a C program to sort the scores.
3	CO2 CO3	A small warehouse tracks how many units of different products are shipped from multiple branches. Another dataset shows how much revenue each product generates per unit. Develop a C program which combines these datasets to calculate the total revenue generated by each branch.
4	CO2 CO3	A basic mobile contact manager stores first and last names separately. For displaying full names in the contact list, you need to join them manually. Additionally, the system must check the length of each full name to ensure It fits the screen. Perform these operations by developing a C program without using built- in string functions.
5	CO2 CO3	A currency exchange booth allows users to convert between two currencies. Before confirming the exchange, the system simulates a swap of the values to preview the result without actually changing the original data. In other cases, it updates the actual values. Develop a C program that implements both behaviours using Call by Value and Call by reference
6	CO2 CO3	A local library needs to store and display detail so fits books ,including title, author ,and year of publication. Design a structure that can hold these details and develop a C program to display a list of all books entered.

Text book:

1. Hassan A fyouni, BehrouzA. Forouzan."A Structured Programming Approach in C",4thEdition, Cengage.

Reference books:

1. Schildt, Herbert."C the complete reference",4thEdition,McGrawHill. Brian W.Kernighan and DennisM. Ritchie,The'C' Programming Language, 2nd edition, Prentice Hallof India.

Online Resources:

1. Introduction to Programming in C [https://onlinecourses.nptel.ac.in/noc23_cs02/preview]
2. C for Everyone: Programming Fundamentals [https://www.coursera.org/learn/c-for-everyone]
3. Computer Programming Virtual Lab[https://cse02-iiith.vlabs.ac.in/exp/pointers/]

4. C Programming: The ultimate way to learn the fundamentals of the C language [https://www.pdfdrive.com/c-programming-the-ultimate-way-to-learn-the-fundamentals-of-the-c-language-e187584209.html]

5. C Programming: The Complete Reference [https://viden.io/knowledge/programming-in-c-language/attachment/28313/c-the-complete-reference-herbert-schildt-4th-edition-pdf/preview]

Code: 1BPOPL107B/109B/207B/209B **Course: Python Programming Lab**

Credits: 1 **L:T:P:S- 0:0:2:0**

SEE: 50 Marks **CIE:50 Marks**

SEE Hours :3 **Total Marks:100**

Prerequisites if any	Basic Knowledge – Familiarity with fundamentals of Python programming(data types, operators, input/output, and program structure). Practical Skills – Ability to use a Python compiler/IDE (write, compile, run programs) and apply basic mathematical and logical reasoning.
Learning objectives	-To build problem-solving skills by developing algorithms and translating them into efficient Python programs. -To gain hands-on experience in writing, compiling, debugging, and executing Python programs for real-world applications.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Develop programs in Python to solve computational problems.	Apply
CO2	Implement Python language programs for real-world applications.	Analyze
CO3	Design and develop Python programs using OOP features to ensure data abstraction, code reusability, and maintainability.	Create

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	2	2	1	1	1	1	1	1	3	2
CO2	3	3	3	2	2	1	1	1	2	2	1	3	3
CO3	2	2	3	2	1	1	1	1	2	3	1	2	2

Mapping Strength: Strong–3 Medium–2 Low–1

List of Experiments

Sl.No	CO	Experiment
1	CO1	a. Develop a program to read the student details like Name, USN, and Marks in three subjects. Display student details, total marks, and percentage with suitable messages. b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.
2	CO1	a. Develop a program to generate Fibonacci sequence of length (N).Read N from the console. b. Write a function to calculate factorial of a number. Develop a program to compute binomial coefficient (Given N and R).
3	CO1	Read N numbers from the console and create a list. Develop a program to print mean, variance, and standard deviation with suitable messages.
4	CO2	Read a multidigit number(as chars)from the console. Develop a program to print the frequency of each digit with suitable message.
5	CO2	Develop a program to print 10 most frequently appearing words in a text file. [Hint: Use dictionary with distinct word sand their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display dictionary slice of first 10 items]
6	CO2	Develop a program to sort the contents of a textfile and write the sorted contents into a separate text file. [Hint: Use string methods strip(), len(), list methods sort(), append(), and file methods open(), read lines(), and write()].
7	CO2	Develop a program to backing Up a given Folder(Folder in a current working directory) into a ZIP File by using relevant modules and suitable methods.
8	CO2	Write a function named Div Exp which takes TWO parameters a, b and returns a value c (c=a/b),raise an exception when b=0.Develop a suitable program which reads two values from the console and calls a function Div Exp.
9	CO3	Define a function which takes TWO objects representing complex numbers and returns new complex number with a addition of two complex numbers. Define a suitable class Complex to represent the complex number. Develop a program to read N (N >=2)complex numbers and to compute the addition of N complex numbers.
10	CO3	Develop a program that uses class Student which prompts the user to enter marks in three subjects and calculates total marks, percentage and displays the score card details. [Hint: Use list to store the marks in three subjects and total marks. Use __init_() method to initialize name, USN and the lists to store marks and total, Use getMarks() method to read marks into the list, and display() method to display the score card details.]

Text books:

1. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers- How to think like a computer scientist: learning with python 3. Green Tea Press, Wellesley, Massachusetts,2020 https://media.readthedocs.org/pdf/howtothink/latest/howtothink.pdf

2. Al Sweigart," Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners",2nd Edition, No Starch Press, 2022. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)

Reference books

1. Kyla McMullen, Elizabeth Matthews and June Jamrich Parsons, Programming with Python, Cengage, 2023.

Web links and Video Lectures (e-Resources):

https://www.learnbyexample.org/python/

https://www.learnpython.org/

https://pythontutor.com/visualize.html#mode=edit

Code: 1BAIA103/203 **Course: Introduction to AI and Applications**

Credits: 3 **CIE: 50 Marks**

L:T:P - 3:0:0 **SEE: 50 Marks**

SEE Hours: 3 **Total Marks:100**

Prerequisites if any	NIL
Learning objectives	To understand the fundamentals of Artificial Intelligence, its techniques, and real-world applications. To develop the ability to apply AI methods for problem-solving while analyzing their societal, ethical, and industrial implications.

Course Outcomes:

On the successful completion of the course, the student will be able to

Co's	Course Outcome	Blooms Level
CO1	Explain the fundamental concepts, types, and working principles of Artificial Intelligence.	Understand
CO2	Illustrate real-world applications of Artificial Intelligence across multiple domains such as healthcare, finance, and education.	Understand
CO3	Apply basic machine learning techniques such as regression, classification, and clustering to solve simple problems.	Apply
CO4	Use prompt engineering techniques to design and execute effective prompts for generative AI tools.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11		PSO1	PSO2
CO1	3	2	1	1	2	1	-	-	-	1	2		2	3
CO2	2	2	2	1	1	3	2	1	1	2	2		2	2
CO3	3	3	3	2	2	1	-	-	2	1	2		3	3
CO4	2	2	3	1	3	1	-	1	2	2	2		3	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module – 1			
1.1	**Introduction to Artificial Intelligence:** Artificial Intelligence, How Does AI Work?, Advantages and Disadvantages of Artificial Intelligence.	2	-	-
1.2	Types of Artificial Intelligence, Weak AI, Strong AI, Reactive Machines, Limited Memory, Theory of Mind, Self-Awareness, Is Artificial Intelligence Same as Augmented Intelligence and Cognitive Computing, Machine Learning and Deep Learning.	2	-	-
1.3	**Machine Intelligence:** Defining Intelligence, Components of Intelligence, Differences Between Human and Machine Intelligence, Agent and Environment.	2	-	-
1.4	**Knowledge Representation:** Introduction, Knowledge Representation, Knowledge-Based Agent, Types of Knowledge.	2	-	-
	Module – 2			
2.1	**Introduction to Prompt Engineering**, Introduction to Prompt Engineering, The Evolution of Prompt Engineering, Types of Prompts, How Does Prompt Engineering Work?, Comprehending Prompt Engineering's Function in Communication, The Advantages of Prompt Engineering, The Future of LLM Communication.	2	-	-
2.2	**Prompt Engineering Techniques for ChatGPT**, Introduction to Prompt Engineering Techniques, Instructions Prompt Technique, Zero, One, and Few Shot Prompting, Self-Consistency Prompt.	2	-	-
2.3	**Prompts for Creative Thinking:** Introduction, Unlocking Imagination and Innovation.	2	-	-
2.4	**Prompts for Effective Writing:** Introduction, Igniting the Writing Process with Prompts.	2	-	-
	Module – 3			
3.1	**Machine Learning:** Techniques in AI.	2	-	-
3.2	Machine Learning Model, Regression Analysis in Machine Learning	2	-	-
3.3	Classification Techniques, Clustering Techniques	2	-	-
3.4	Naïve Bayes Classification, Neural Network, Support Vector Machine (SVM).	2	-	-
	Module – 4			
4.1	**Trends in AI:** AI and Ethical Concerns, Recent trends in AI.	2	-	-
4.2	AI as a Service (AIaaS)	2	-	-
4.3	Expert System	2	-	-
4.4	Internet of Things, Artificial Intelligence of Things (AIoT).	2	-	-
	Module – 5			
5.1	**Industrial Case of AI:** Case study of AI in Healthcare	2	-	-
5.2	Case study of AI in Finance, Case study of AI in Retail	2	-	-
5.3	Case study of AI in Agriculture, Case study of AI in Education,	2	-	-

5.4	Case study of AI in Transportation, AI in Experimentation and Multi-disciplinary research.	2	-	-
	Total No. of Lecture Hours	**40**	-	-
		Total No. of Tutorial Hours	**00**	-
			Total No. of Practical Hours	**0**

Suggested Learning Resources: (Textbook/ Reference Book/ Manuals):

Text books:

1. Reema Thareja, Artificial Intelligence: Beyond Classical AI, Pearson Education, 2023.
2. Ajantha Devi Vairamani and Anand Nayyar, Prompt Engineering: Empowering Communication, 1st Edition, CRC Press, Taylor & Francis Group, 2024. (DOI: https://doi.org/10.1201/9781032692319).
3. Saptarsi Goswami, Amit Kumar Das and Amlan Chakrabarti, "AI for Everyone – A Beginner's Handbook for Artificial Intelligence", Pearson, 2024.

Reference books / Manuals:

1. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach* (4th Edition), Pearson Education, 2023.
2. Elaine Rich, Kevin Knight, and Shivashankar B. Nair, *Artificial Intelligence*, McGraw Hill Education.
3. Tom Taulli, *Prompt Engineering for Generative AI: ChatGPT, LLMs, and Beyond*, Apress, Springer Nature.
4. Nilakshi Jain, Artificial Intelligence: Making A System Intelligent, First Edition, Wiley.

Web links and Video Lectures (e-Resources):

1. Elements of AI – *https://www.elementsofai.com*
2. CS50's Introduction to Artificial Intelligence with Python – Harvard *https://cs50.harvard.edu/ai/*
3. Google Machine Learning Crash Course – *https://developers.google.com/machine-learning/crash-course*
   Learn Prompting (Open-Source Guide) – *https://learnprompting.org*
4. Google AI – Learn with Google AI *https://ai.google/education/*
5. Coursera – Machine Learning by Andrew Ng (Stanford University) *https://www.coursera.org/learn/machine-learning*
6. OpenAI Prompt Engineering Guide (for ChatGPT) *https://platform.openai.com/docs/guides/gpt-best-practices*
7. Prompt Engineering for Developers – DeepLearning.AI + OpenAI *https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/*
8. Ethics in AI – Google Responsible AI Practices *https://ai.google/responsibilities/responsible-ai-practices/*

Google Teachable Machine (Train AI models visually without code) https://teachablemachine.withgoogle.com

Teaching-Learning Process (Innovative Delivery Methods):
The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching-learning process and facilitate the achievement of course outcomes.

- Flipped Classroom
- Problem-Based Learning (PBL)
- Case-Based Teaching
- Simulation and Virtual Labs
- ICT-Enabled Teaching
- Tool Demonstration

Assessment Structure:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage.

- To qualify and become eligible to appear for SEE, in the **CIE**, a student must score at least **40% of 50 marks**, i.e., **20 marks**.

- To pass the **SEE**, a student must score at least **35% of 50 marks**, i.e., **18 marks**.

  Notwithstanding the above, a student is considered to have **passed the course,** provided the combined total of **CIE and SEE is at least 40 out of 100 marks**

Continuous Comprehensive Assessments (CCA):

CCA will be conducted for a total of 25 marks. It is recommended to include a maximum of two learning activities aimed at enhancing the holistic development of students. These activities should align with course outcomes and promote higher-order thinking and application-based learning.

Learning Activity -1: Practical Assignment on Creating Effective Prompts (Marks- 25)

INSTRUCTIONS:

1. Students must demonstrate the solutions to the course instructor and submit the record containing prompt creation (procedure), prompt execution and results with observations.

2. Course instructor must evaluate the student performance as per the rubrics.

Code: 1BPLC205E/105E **Course: Introduction to C Programming**

Credits: 4 **L: T:P – 3:0:0**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	NIL
Learning objectives	To acquire fundamental programming concepts, methodologies, and structures that are essential for developing efficient and reliable C programs.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	**Course Outcomes**	**Bloom's level**
CO1	Explain the fundamental structure of a C program and primitive constructs.	Understanding
CO2	Apply decision making and iterative control structures to solve computational problems.	Apply
CO3	Develop programs using arrays, functions and string operations to solve real-world problems.	Apply
CO4	Implement structures and pointers for complex data representation.	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11		PSO1	PSO2	PSO3
CO1	3	2	0	0	0	-	-	0	2	-	1		Department Specific		
CO2	3	3	3	2	2	-	-	1	2	-	2				
CO3	3	3	3	3	3	-	-	2	2	-	2				
CO4	3	3	3	3	3	-	-	2	2	-	2				

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

No.	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Module 1				
1.1	**Overview of C:** History of C, Importance of C, Basic Structure of C Programs	1		
1.2	Programming Style, Compiling and Executing a 'C' Program.	1		
1.3	**Constants, Variables and Data Types:** Character Set, C Tokens	1		
1.4	Keywords and Identifiers, Constants, Variables,	1		
1.5	Data Types, Declaration of Variables,	1		
1.6	Assigning Values to Variables, Defining Symbolic Constants	1		
1.7	Declaring a Variables as Constants and Volatile	1		
1.8	Input/ Output Statements in C	1		
Module 2				
2.1	Operators: Introduction to Operators, Arithmetic Operators	1		
2.2	Relational Operators, Logical Operators, Assignment Operators,	1		
2.3	Decrement Operators, Conditional Operators,	1		
2.4	Precedence of Arithmetic Operators. Introduction, Decision Making with IF Statement,	1		
2.5	Simple IF Statement, The IF-ELSE Statement, Nesting of IF-ELSE Statements	1		
2.6	The ELSE IF Ladder, The Switch Statement	1		
2.7	The ?: Operator, The GOTO Statement	1		
2.8	WHILE, DO, FOR, Jumps in LOOPS	1		
Module 3				
3.1	Arrays and Strings: Introduction, Declaration and Initialization of One-dimensional and Two-Dimensional Arrays	1		
3.2	Declaring and Initializing String Variables	1		
3.3	Example programs using arrays	1		

3.4	Reading Strings from Terminal	1		
3.5	Writing Strings to Screen	1		
3.6	Arithmetic Operations on Characters	1		
3.7	Comparison of Two Strings	1		
3.8	String-handling Functions	1		
Module 4				
4.1	User-defined Functions: Introduction, Need for User-defined Functions	1		
4.2	A Multi-functional Program, Elements of User-defined Functions,	1		
4.3	Definition of Function, ,	1		
4.4	Return Values and their Types	1		
4.5	Function Calls, Function Declaration	1		
4.6	No Arguments and no Return Values	1		
4.7	Arguments but no Return Values	1		
4.8	Nesting of Functions.	1		
Module 5				
5.1	Structures and Pointers: Introduction	1		
5.2	Defining a Structure & Declaring and Accessing Structure Variables and Members,	1		
5.3	Structure Initialization and Copying and Comparing Structure Variables	1		
5.4	Array of Structures, Arrays within Structures.	1		
5.5	Pointers: Introduction, Understanding Pointers	1		
5.6	Accessing the Address of Variable	1		
5.7	Declaring pointer variables, initialization of pointers,	1		
5.8	accessing variables through its pointer	1		
	Total No. of Lecture Hours	40	-	-
	Total No. of Tutorial Hours		00	-
	Total No. of Practical Hours			10

Textbook:

1. Programming in ANSIC,9e, E Balaguru swamy, Tata McGraw Hill Education.

Reference Book:

1. PROGRAMMING IN C, Reema Thareja, Oxford University, Third Edition,2023.
2. The ‘C’ Programming Language, Brian W. Kernighan and Dennis M. Ritchie, Second Edition, Prentice Hall of India, 2015

Online Resources:

1. E learning.vtu.ac.in/econtent/courses/video/BS/15PCD23.html
2. https://nptel.ac.in/courses/106/105/106105171/MOOC

Code: 1BPOPL107E/109E/207E/209E **Course: Programming in C Lab**

Credits: 1 **L:T:P –** 0:0:2:0

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	
Learning objectives	• To develop problem-solving skills through the implementation of fundamental programming constructs such as loops, functions, arrays, strings, pointers, and structures. • To apply modular programming concepts in solving real-world problems using C language.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Apply basic programming constructs such as input/output, decision making, loops, arrays, and functions to solve mathematical and logical problem.	Apply
CO2	Apply modular programming techniques using strings, pointers, and structures to develop programs for real-life applications	Apply
CO3	Analyze problem statements and design structured solutions using arrays, matrices, and modular programming concepts	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11		PSO1	PSO2
CO1	3	2	2	2	3	1	-	1	2	1	2		3	2
CO2	3	2	3	2	3	1	-	2	2	1	2		3	3
CO3	3	3	3	3	3	2	1	2	2	2	3		3	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

List of Experiments

Sl.No	CO	Experiment
PART – A CONVENTIONAL EXPERIMENTS		
1	CO1	a) Develop a program to calculate the temperature converter from degree to Fahrenheit. b) Develop a program to find the roots of quadratic equations
2	CO1	a) Develop a program to find whether a given number is prime or not. b) Develop a program to find key elements in an array using linear search.
3	CO1	Given age and gender of a person, develop a program to categories senior citizen (male& female).
4	CO1, CO3	a) Generate Floyd's triangle for given rows. b) Develop a program to find the transpose of a matrix.
5	CO2	Develop a program to concatenate two strings, find length of a string and copy one string to other using string operations
6	CO2	Develop a modular program to find GCD and LCM of give numbers.
7	CO2	Develop a program to declare the structure of employees and display the employee records with higher salary among two employees.
8	CO2, CO1	a) Develop a program to add two numbers using the pointers to the variables. b) Develop a program to find the sum of digits of a give number.
9	CO3	Develop a program to perform matrix Multiplication.
10	CO2	Develop a program to create an array of structures to store book details and check whether a specific book, as requested by the user, is available or not.

Text book:

1. Programming in ANSIC,9e, E Balaguru swamy, Tata McGraw Hill Education

Reference books:

1. PROGRAMMING IN C, Reema Thareja, Oxford University, Third Edition,2023.
2. The 'C' Programming Language, Brian W. Kernighan and Dennis M. Ritchie, Second Edition, Prentice Hall of India, 2015

Online Resources:

1. Introduction to Programming in C [https://onlinecourses.nptel.ac.in/noc23_cs02/preview]
2. C for Everyone: Programming Fundamentals [https://www.coursera.org/learn/c-for-everyone]
3. Computer Programming Virtual Lab[https://cse02-iiith.vlabs.ac.in/exp/pointers/]
4. C Programming: The ultimate way to learn the fundamentals of the C language [https://www.pdfdrive.com/c-programming-the-ultimate-way-to-learn-the- \fundamentals-of-the-c- language-e187584209.html
5. C Programming :The Complete Reference [https://viden.io/knowledge/programming-in-c-language/attachment/28313/c-the-complete-reference-herbert-schildt-4th-edition-pdf/preview]
6. E learning.vtu.ac.in/econtent/courses/video/BS/15PCD23.html

Code: BESC104E/204E **Course: Essentials of Information Technology**

Credits: 3 **L: T:P – 3:0:0**

SEE: 50 Marks **CIE: 50 Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	NIL
Learning objectives	• Gain foundational knowledge of computer systems, software, networking, and data management. • Develop skills in problem-solving, communication, and ethical IT practices.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Explain fundamental concepts of Information Technology	Understand
CO2	Describe the role of cyber security and ethics issues in Information Technology	Understand
CO3	Apply basic software engineering concepts for Website and application development.	Apply
CO4	Develop queries for quick insert, access and updating of structured information.	Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11		PSO1	PSO2
CO1	3	1	1						1	2	3		1	1
CO2	3	1	1		3	3			2	2	3			1
CO3	3	3	3	2	3				2	2	3		2	2
CO4	3	3	3	2	2				2	2	2		3	2

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structure

No.	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
	Module 1			
1.1	**Data Storage:** Bits and Their Storage	1		
1.2	Main Memory, Mass Storage	1		
1.3	Representing Information as Bit Patterns, The Binary System	1		
1.4	Storing Integers, Storing Fractions	1		
1.5	**Data Manipulation:** Computer Architecture	1		
1.6	Machine Language, Program Execution,	1		
1.7	Arithmetic/Logic Instructions,	1		
1.8	Communicating with Other Devices.	1		
	Module 2			
2.1	**Operating Systems:** The History of Operating Systems	1		
2.2	Operating System Architecture,	1		
2.3	Coordinating the Machine's Activities	1		
2.4	Handling Competition Among Processes, Security.	1		
2.5	**Algorithms:** The Concept of an Algorithm	2		
2.6	Algorithm Representation	1		
2.7	Algorithm Discovery.	1		
	Module 3			
3.1	**Networking and the Internet:** Network Fundamentals, The Internet, **Ethical Issues in Information Technology:** Overview, Ownership Rules, Ethics and Online Content.	1		
3.2	The World Wide Web, Internet Protocols, Security.	1		

3.3	**Cybersecurity:** Overview—What is Cybersecurity?, Brief History of Cybersecurity Events,	1		
3.4	The Basic Information Security Model	1		
3.5	Cyber Hygiene, Teams in Cybersecurity.	1		
3.6	Ethical Issues in Information Technology	1		
3.7	Ownership Rules	1		
3.8	Ethics and Online Content.	1		
	Module 4			
4.1	**Software Engineering:** The Software Engineering Discipline	1		
4.2	The Software Life Cycle	1		
4.3	Software Engineering Methodologies	1		
4.4	Modularity	1		
4.5	Tools of the Trade	1		
4.6	Database Systems	1		
4.7	Database Fundamentals	1		
4.8	The Relational Model	1		
	Module 5			
5.1	**Introduction to HTML and Website Development:** What is HTML?	1		
5.2	Cascading Style Sheets (CSS)	2		
5.3	Website Design and Storyboarding	1		
5.4	Structure of a Website.	1		
5.5	**Computer Graphics:** The Scope of Computer Graphics	1		
5.6	Modeling	1		
5.7	Rendering	1		
	Total No. of Lecture Hours	40	-	-
	Total No. of Tutorial Hours		00	-
	Total No. of Practical Hours			00

Text book:

1. J. Glenn Brookshear and Dennis Brylow, Computer Science: An Overview, 12th Edition, Pearson Education Limited, 2017.
2. Roy, Shambhavi; Daniel, Clinton; and Agrawal, Manish, "Fundamentals of Information Technology", Digital Commons at The University of South Florida (2023). https://digitalcommons.usf.edu/dit_tb_eng/19

Reference Book:

1. V. Rajaraman, “Introduction to Information Technology”, Third Edition, PHI Learning, 2018.
2. Pelin Aksoy, Information Technology in Theory, First Edition, Cengage.

Web links and Video Lectures (e-Resources):

- Information Technology: https://onlinecourses.swayam2.ac.in/cec20_cs05/preview
- Computer Organization and Architecture: https://nptel.ac.in/courses/106103068
- Introduction To Internet: https://nptel.ac.in/courses/106105084

** *** **